



Tipos de datos

Transcripción

[00:00] Hola, alumnos y alumnas, una vez más les doy la bienvenida para nuestra aula número 3, en la cual estaremos hablando sobre la administración de las tablas en las bases de datos, e inicialmente queremos también hablar un poco sobre los tipos de datos que podemos tener en nuestros campos.

[00:19] Entonces para crear una tabla debemos establecer no solamente el nombre de los campos sino el tipo de datos que vamos a almacenar en esos campos. Entonces vamos a iniciar con los números enteros. Tengo las siguientes posibilidades: tengo tinyint, smallint, mediumint, int y bigint. Cada uno de ellos tiene un valor de bytes.

[00:40] Recordemos que un byte son 8 bits, entonces un byte tinyint almacenaría 8 bits, lo cual me permite un máximo de 255 valores. Esto cuando no tiene signo. Recordemos que esta propiedad unsigned quiere decir sin signo. Entonces iría de 0 a 255. Ahora bien, si yo tengo signed, o sea colocando signo, puedo ir desde -128 hasta 127 con smallint.

[01:16] Entonces el valor en bytes sería 2 bytes, o sea 16 bits. Entonces, menor valor cuando tiene signo es -32768 y mayor valor cuando tiene signo 32767, el mayor valor sería 2 a la 16 -1, que sería 65535 el número. Iría de cero a 65535 y así en adelante.

[01:44] Mediumint entonces también serían 3 bytes, y estos son el menor valor teniendo signo, el menor valor sin signo y el mayor valor sin signo sería 16777215. Bueno, no les voy a hablar específicamente de los valores, pero pa

que ustedes me entiendan de acuerdo al tipo de entero que utilicemos, asimismo también será la capacidad del mismo.

[02:16] Entonces, por ejemplo este bigint tiene 2 elevado a la 64-1. La capacidad es un número gigante, ni siquiera lo podemos escribir aquí. Entonces importante recordar, la propiedad unsigned quiere decir sin signo. Entonces avancemos. También podemos tener números decimales, existen unos números decimales que pueden ser variables, que sería float y double.

[02:44] Float es el número que tiene coma flotante, esto quiere decir que si nosotros declaramos un float 64, significa que él tiene en total 6 dígitos, de los cuales 4 van a ser decimales. Entonces si nosotros incluimos el valor 76.00009, entonces, él va a almacenar 76.0001, ya redondea esta cifra.

[03:13] Entonces el float tiene una precisión simple de hasta 4 bytes y el double tiene una precisión exacta. Si queremos un número de mucha precisión en nuestros datos usamos doble, si necesitamos precisión pero no tanto, usamos float.

[03:30] Ahora tenemos también los números decimales fijos, que sería el tipo decimal o numérico, que tiene una cantidad máxima de dígitos de hasta 65. Entonces, si yo declaro un decimal de cinco números, de los cuales tres van a ser casillas decimales, entonces solo podremos almacenar desde -99.999 hasta 99.999.

[03:58] Aquí si tiene signo o no tiene el signo, no afecta la aplicación de esta variable. También tenemos el bit propiamente dicho, entonces el tipo bit puede almacenar hasta 64 bits, entonces estamos hablando de la misma capacidad que vimos en los enteros de 2 elevado a la 64, de 2 por 10 a la 64.

[04:21] Entonces aquí, por ejemplo, si yo tengo bit igual a 1, puede ser 0 o 1. En un campo bit yo no puedo introducir otro valor, no puedo introducir 2 ni 3 ni nada por el estilo, y un bit 3, entonces tengo las combinaciones de binarios

para quienes conocen un poco de la tabla de binarios, ustedes saben que este es el 0, este es el 1, el 2, el 3, el 4, el 5, el 6 y el 7.

[04:44] Avancemos. Ahora, estos campos numéricos tienen los siguientes atributos, la propiedad signed o unsigned, con signo o sin signo, como ya lo hablamos. Zeropfill llena los espacios con cero. Entonces, por ejemplo, si yo tengo declarado un entero de 5 y se lo almaceno con esa propiedad de zerofill, entonces ese 54 va a quedar escrito 00054.

[05:10] El problema con zerofill es que ocupa espacio de memoria para almacenar. Entonces hay casos en los cuales, sí es necesario que tengas zerofill, hay otros casos en los cuales no. Auto_increment: entonces, existe un incremento secuencial, es un campo que va a aumentar.

[05:27] Por ejemplo, se crea, no hay que hacer más nada y él solito va aumentando en el rango, en el intervalo de rangos que nosotros definamos: de 2 en 2, de 5 en 5, de 10 en 10, de 1000 en 1000, lo que ustedes quieran. Entonces aquí vemos ejemplos.

[05:41] También tenemos un error que puede suceder que es out of range, que se presenta cuando los valores se salen de estos límites establecidos. Entonces si nosotros declaramos un campo int 5 y de repente, digitamos 1350824, ya tenemos 7 dígitos. Entonces, ahí ya va aparecernos error out of range porque el campo no fue declarado así.

[06:08] Bien. Ahora también tenemos el tipo de datos de fecha y hora. Tenemos date que es el más usado en SQL, que tiene el siguiente rango, viene desde el primero de enero del año 1000 hasta el 31 de diciembre del año 9999. El orden de él es: año, mes, día. Ese es el formato y así es como vamos a trabajar en nuestras tablas.

[06:35] Más adelante, cuando realicemos nuestros ejercicios, ustedes van a ver que mantenemos ese formato, año, mes, día. Siguiendo. Está el datetime, que él ya no solamente nos almacena la fecha, sino que también nos almacena la

hora. Este también es muy útil, sobre todo cuando se está trabajando con logs, que necesitamos saber en qué momento se hizo una actualización, en qué momento algún usuario modifica un archivo, entonces esto es muy importante.

[07:05] Timestamp, él ya tiene en consideración el cambio horario de cada país de UTC, universal time coordinated. Entonces él tiene un rango inferior porque pues realmente no necesita tener algo mayor. ¿A qué me refiero?

[07:21] Que si, por ejemplo vamos a programar una reunión, por ejemplo, estamos aquí en Brasil, digamos actualmente, pues yo estoy en Brasil y si vamos a programar una reunión con China, entonces pues lógicamente tenemos que hallar un horario en el cual podamos encontrarnos, que todo quede bien coordinado.

[07:40] Y no digamos por ejemplo son las 21:00 de la noche, aquí y allá es las 9:00 de la mañana el otro día, bueno, se podría hacer así, pero si la persona no sabe, la persona se va a reunir allá en un horario que no coincide, entonces esa es la idea. ¿Por qué hasta el 2038? Pues nadie va a programar una reunión, para el 2038, desde ahorita, digamos, esto es algo que tal vez solo se ve en las películas.

[08:04] Siguiente. El tipo time, almacena únicamente como las horas. Entonces, pues digamos realmente el día tiene 24 horas, entonces tener un rango de 838 a 800, de -838 a 839, pues tal vez no tenga mucha utilidad, para eso se usa datetime y de repente se seleccionan las horas que se necesitan. Pero bueno, es un campo que existe, no vamos a entrar a polemizar sobre él.

[08:36] Está también year, que puede expresarse en formato de dos o cuatro dígitos y cuando se expresa en formato de dos dígitos, pues los separa como dividiendo un decimal, pero pues también no tiene mucha aplicación únicamente hacerlo así. Para eso existe date, o sea, date es muy versátil.

[08:55] Ahora vamos a hablar un poco del tipo de variables string. Tenemos primero char o char, como quieran pronunciarlo, char en inglés, me imagino porque viene de character. Entonces, pero hablemos char. Cadena de caracteres con valor fijo de 0 a 255. Entonces, por ejemplo, si yo tengo un char de 4, entonces voy a almacenar, sí quiero almacenar "aa", me va a cenar "***aa" con dos espacios.

[09:23] Estos dos punticos representan dos espacios. Entonces ocupa más memoria, me va a ocupar cuatro espacios de memoria siendo que solamente estoy usando dos. Y por eso existe también varchar, que es una cadena de caracteres con un valor variable también de 0 a 255, pero la diferencia de varchar es que él varia si yo escribo "aa", entonces él va a guardar únicamente "aa" y reserva el otro espacio de memoria, no lo va a utilizar. Entonces es lo que más se utiliza.

[09:50] De hecho, cuando creamos nuestras bases de datos, nuestras tablas, ustedes van a notar que lo que más usamos, es varchar. Binary es una cadena de caracteres con valor fijo de 0 a 255 con números binarios, o sea, trabaja con bits ceros y unos. Se usa bastante también.

[10:08] Varbinary, la misma idea varchar, él usa pues lo que necesita. Entonces ya saben, tienen como el mismo principio, son análogos a char y varchar. Blob, binary long. Entonces, podemos querer almacenar, por ejemplo, una foto o un formato grande que está en binario, entonces queremos tener la oportunidad de guardarlos pues de acuerdo con la necesidad: tinyblob, mediumblob, longblob.

[10:44] Text, cuando queremos almacenar un texto largo, por ejemplo, una secuencia de palabras, entonces tenemos tinytext, mediumtext, longtext. Enum. Entonces nos permite definir opciones en una lista predefinida, valga la redundancia. Entonces, por ejemplo, si yo creo un campo talla con enum, entonces yo ya le estoy diciendo a mi campo que allí dentro puedo tener las tres opciones siguientes o pequeño o medio o grande.

[11:14] Esto es muy útil. Entonces digamos, lo limitamos únicamente a estas tres opciones dentro de la lista. Entonces esto, con respecto a los strings. Los atributos de los strings, set y collate, que es algo que vimos previamente cuando estábamos creando nuestras tablas, que es el tipo o el conjunto de caracteres que vamos a utilizar, si es UTF-8 UTF-16, si son ideogramas japoneses, si es el alfabeto ruso, en fin, el alfabeto árabe.

[11:47] Esto nos ayuda a nosotros a establecer el tipo de tabla ASCII que estaremos trabajando, el tipo de caracteres ASCII que estaremos trabajando. También finalmente, tenemos los campos espaciales, hoy en día todos quieren saber de mapas, hoy en día muchos aplicativos e Internet trabajan con localización espacial. Siempre nos preguntan que quieren saber nuestra localización, etcétera.

[12:12] Ustedes ya saben de lo que estoy hablando. Entonces aquí yo puse pues GPS, digamos, global positional system, claro, lógicamente, pues hay que almacenar coordenadas, entonces tenemos las siguientes opciones: tenemos point, que es el más utilizado porque almacena la coordenada exacta, entonces un punto geográfico lo almacena tal cual y ahí necesitamos toda la precisión posible del caso.

[12:35] Entonces, muy útil. Geometry y polygon son campos espaciales que se relacionan con el área y linestring es como una línea. Entonces en una dimensión geometry y polygon dos dimensiones, y point es un punto específico.

[12:51] Entonces básicamente esto es de manera general tipos de variables que podemos trabajar en SQL y encuentran otro tipo de variables, pero son tipos de datos que tal vez no usemos. Sin embargo, esto es como lo principal, que conozcamos en cuanto a tipos de datos. Entonces, hasta el próximo video.